UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/044,731	01/11/2002	Raj Prakash	SP-6540 US	1596

22120 7590 05/06/2005

ZAGORIN O'BRIEN GRAHAM LLP
7600B N. CAPITAL OF TEXAS HWY.
SUITE 350
AUSTIN, TX 78731

EXAMINER	
MITCHELL, JASON D	

ART UNIT	PAPER NUMBER
2193	

DATE MAILED: 05/06/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	10/044,731	PRAKASH, RAJ	
	Examiner	Art Unit	
	Jason Mitchell	2193	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *17 February 2005*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-8 and 33-48* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-8 and 33-48* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *17 February 2005* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

This action is in response to an application filed on 1/11/2002.

As per Applicant's request claims 1-3 have been amended, claims 9-32 have been canceled, and claims 33-48 have been added. Claims 1-8 and 33-48 are pending in this case.

Claim Rejections - 35 USC § 112

The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

Claim 1 recites the limitation "the infrequent null pointer condition check" in line 5. There is insufficient antecedent basis for this limitation in the claim.

Examiner assumes Applicant intended to delete 'infrequent' as was done in lines 1 and 3. Consequently for the purposes of this examination the claim will be treated as if line 5 also read 'the null pointer condition check'.

Claim 4 is rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. The claim recites 'passing fault to target translation data from the fault to target translation table to the compiler using a handler program'. There is no disclosure as to what data is being passed, or when and why it is being passed, therefore it is unclear what the sited claims encompass.

Because of the similarity between claim 4 and original claim 3, claim 4 will be treated as reciting the same limitations as amended claim 3 for the purposes of this action.

Claim 45 recites the limitation "the associations" in line 2. There is insufficient antecedent basis for this limitation in the claim.

Claim 46 recites the limitations "the null pointer condition check dependent program code" and "the null pointer condition handling program code" in lines 2 and 3 respectively. There is insufficient antecedent basis for these limitations in the claim.

Claim Rejections - 35 USC § 101

35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

Claims 44-48 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

Claims 44-48 are not limited to tangible embodiments. In view of Applicant's disclosure, specification page 9, lines 31-32 'Additionally, applications may be in the form of electronic signals', the medium is not limited to tangible embodiments, instead being defined as including both tangible embodiments (e.g., hard disk drive, optical drive) and intangible embodiments (e.g., signals). As such, the claim is not limited to statutory subject matter and is therefore non-statutory.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claims 1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21 and 23 are rejected under 35 U.S.C. 103(a) as being unpatentable over 'Optimizing Away C++ Exception Handling' by Schilling (Schilling) in view of 'Effective Null Pointer Check Elimination Utilizing Hardware Trap' by Kawahito et al. (Kawahito) and further in view of US 6,189,141 to Benitez et al. (Benitez).

Regarding Claim 1: Schilling discloses creating a fault to target translation table (pg. 40, col. 1, par. 5 'building ... tables'), relating the fault condition to a procedural instruction in the fault to target translation table (pg. 40, col. 1, par. 5 'that relate ranges of instruction counter values to ... exception handling'); and compiling the source program to an executable program (pg. 40, col. 1, par. 5 'at compile and link time'). Schilling does not explicitly disclose the fault table handling null pointer conditions, but instead Schilling discloses handling exceptions in general (pg. 40, col. 1, par. 5 'exception handling').

Kawahito teaches eliminating explicit Null Pointer tests (pg. 139, col. 1, par. 3 'no explicit instruction has to be generated to check the null pointer') through use of the Null Pointer Exception (pg. 139, col. 1, par. 3 'accessing the zero address will throw an exception') in an analogous art for the purpose of optimizing the execution of a program (pg. 139, col. 2, par. 6 'converted to hardware traps ... to minimize the execution cost').

Neither Schilling nor Kawahito disclose creating the fault to target translation table on the condition the null pointer condition check infrequently encounters null pointer conditions.

Benitez teaches gathering statistics as to the number of times a path is executed and determining, based on said gathering, when to optimize that path (col. 29, lines 31-33 'if control passes through an arc ... a number of times that is equal to a start-trace threshold, hot trace selector is invoked to select a hot trace'), in an analogous art for the purpose of providing dynamic optimization (col. 32, lines 26-27 'dynamically optimizes hot trace').

It would have been obvious to a person of ordinary skill in the art at the time of the invention to populate the exception table disclosed in Schilling (pg. 40, col. 1, par. 5 'building ... tables') with the null pointer exceptions disclosed in Kawahito (pg. 139, col. 1, par. 3 'accessing the zero address will throw an exception'), because one of ordinary skill in the art would have been motivated to handle exceptions thrown by null pointer references thereby providing program optimization (pg. 139, col. 2, par. 6 'converted to hardware traps ... to minimize the execution cost').

Further, it would have been obvious to a person of ordinary skill in the art at the time of the invention to use Benitez's hot trace designation (col. 29, lines 31-33 'hot trace selector') to only implement the null pointer test elimination taught by Kawahito (pg. 139, col. 2, par. 6 'converted to hardware traps') when the null pointer condition was infrequent, as determined by a frequent execution of the non-null pointer trace (Benitez col. 29, lines 31-34 'if control passes through an arc of a current hot block a number of

times ... select a hot trace'), because one of ordinary skill in the art would have been motivated to apply the optimizations where they would do the most good (Benitez col. 2, lines 28-31 'selecting sequences ... that are most amenable to dynamic optimization').

Regarding Claim 2: The rejection of claim 1 is incorporated, respectively; further, Schilling and Kawahito do not disclose gathering statistics regarding, and determining an acceptable rate of, occurrences of the infrequent null pointer condition. However Kawahito does disclose his techniques as being applicable to a dynamically compiled language, namely JAVA™ (pg. 139, col. 1, par. 1 'a new algorithm ... written in JAVA™').

Benitez teaches gathering statistics as to the number of times a path is executed and determining, based on said gathering, when to optimize that path (col. 29, lines 31-33 'if control passes through an arc ... a number of times that is equal to a start-trace threshold, hot trace selector is invoked to select a hot trace'), in an analogous art for the purpose of providing dynamic optimization (col. 32, lines 26-27 'dynamically optimizes hot trace').

It would have been obvious to a person of ordinary skill in the art at the time of the invention to use Benitez's hot trace designation (col. 29, lines 31-33 'hot trace selector') to only implement the null pointer test elimination taught by Kawahito (pg. 139, col. 2, par. 6 'converted to hardware traps') when the null pointer condition was infrequent, as determined by a frequent execution of the non-null pointer trace (Benitez col. 29, lines 31-34 'if control passes through an arc of a current hot block a number of times ... select a hot trace'), because one of ordinary skill in the art would have been motivated

to apply the optimizations where they would do the most good (Benitez col. 2, lines 28-31 'selecting sequences ... that are most amenable to dynamic optimization').

Regarding Claim 3: The rejection of claim 1, is incorporated; further, Schilling discloses responsive to a fault that corresponds to a null pointer condition, passing fault to target translation data from the fault to target translation table to the compiler (pg. 40, col. 1 par. 5 'if an exception is thrown ... looks up the current instruction counter in the tables') using a handler program to direct the fault to a target indicated by the fault to target translation data (pg. 40, col. 1 par. 5 C++ runtime system ... give control to a catch handler').

Regarding Claim 4: The rejection of claim 2, is incorporated; further, Schilling discloses responsive to a fault that corresponds to a null pointer condition, passing fault to target translation data from the fault to target translation table to the compiler (pg. 40, col. 1 par. 5 'if an exception is thrown ... looks up the current instruction counter in the tables') using a handler program to direct the fault to a target indicated by the fault to target translation data (pg. 40, col. 1 par. 5 C++ runtime system ... give control to a catch handler').

Regarding Claims 5, 7: The rejection of claims 1, and 3 are incorporated respectively; further Schilling discloses accessing the fault to target translation table (pg. 40, col. 1, par. 5 'building ... tables') during compiling of the source program (pg. 40, col. 1, par. 5 'at compile and link time').

Regarding Claims 6, 8: The rejection of claims 2, and 4 are incorporated respectively; further Schilling discloses accessing the fault to target translation table (pg. 40, col. 1,

par. 5 'building ... tables') during compiling of the source program (pg. 40, col. 1, par. 5 'at compile and link time').

Claims 33-48 are rejected under 35 U.S.C. 103(a) as being unpatentable over US 6,189,141 to Benitez et al. (Benitez) in view of 'Effective Null Pointer Check Elimination Utilizing Hardware Trap' by Kawahito et al. (Kawahito) and further in view of 'Optimizing Away C++ Exception Handling' by Schilling (Schilling).

Regarding Claim 33, 41, and 44: Benitez discloses optimizing code according to profile feedback for the code (col. 31, lines 57-59 'optimizes hot traces that have been selected by selector 204') that indicates a condition occurs more frequently than a given threshold (col. 29, lines 31-34 'if control passes through an arc ... a number of times that is equal to ... threshold') and thus, inherently, discloses optimizing code when a complimentary condition occurs less frequently than the given threshold. However Benitez does not place any limits on the type of optimization applied to the code, instead disclosing use of 'any of a variety of known techniques' (col. 32, line 26). Kawahito teaches a program code optimization technique (pg. 139, col. 2, par. 4 'our optimization') which includes eliminating from code, null pointer condition checks (pg. 139, col. 2, par. 5 'null checks ... are converted to hardware traps'). Kawahito teaches the use of exception handling (pg. 139, col. 2, par. 5 'hardware traps') but does not provide any explicit implementation details regarding said exception handling. Schilling teaches a method of exception handling including installing exception entries in a table (pg. 40, col. 1, par. 5 'building ... tables'), wherein the installed entries are

utilized to direct execution of the code to respective limits of the code that handle the associated exception (pg. 40, col. 1, par. 5 'that relate ranges of instruction counter values to ... exception handling').

It would have been obvious to a person of ordinary skill in the art at the time of the invention to implement a profile based optimization method (col. 31, lines 57-59), as disclosed in Benitez, using the specific null-pointer check optimization taught in Kawahito (pg. 139, col. 2, par. 5), supported by the exception handling taught in Schilling (pg. 40, col. 1, par. 5) because one of ordinary skill in the art would have been motivated to minimize execution costs (Kawahito pg. 139, col. 1, par. 1 'in order to minimize the execution cost') of a dynamically translated program (pg. 139, col. 2, par. 5) by only applying the null-pointer optimization to where it is most needed (Benitez, Abstract 'reduces unnecessary translations and optimizations, and thereby increases execution speed').

Regarding Claim 34: The rejection of claim 22 is incorporated; further Benitez discloses identifying conditions that are encountered more frequently than the given threshold (col. 29, lines 31-34 'if control passes through an arc ... a number of times that is equal to ... threshold') and thus, inherently, discloses identifying conditions when a complimentary condition occurs less frequently than the given threshold.

Regarding Claim 35, 43, 48: The rejection of claim 33 is incorporated; further Benitez discloses extracting information about condition frequencies from the profile feedback for the code (col. 29, lines 31-34 'if control passes through an arc ... a number of times that is equal to ... threshold').

Regarding Claim 36: The rejection of claim 35 is incorporated; further Benitez discloses determining those of the condition checks that have profile feedback information that indicates the condition occurs more frequently than the given threshold (col. 29, lines 31-34 'if control passes through an arc ... a number of times that is equal to ... threshold') and thus, inherently, discloses determining those of the condition checks that have profile feedback information that indicates the complimentary condition occurs less frequently than the given threshold.

Regarding Claim 37, 47: The rejection of claim 33 is incorporated; further Benitez discloses generating executable code from the optimized code (col. 31, lines 57-59 'dynamically translates').

Regarding Claim 38, 42 and 45: The rejection of claim 33 is incorporated; further Schilling teaches generating the table to associate faults with respective exception handling code units (pg. 40, col. 1, par. 5 'that relate ranges of instruction counter values to ... exception handling').

Regarding Claim 39, 46: The rejection of claim 38 is incorporated; further Schilling teaches populating the table with instruction identifiers of instructions associated with the exception conditions and respective ones of the exception handling code units (pg. 40, col. 1, par. 5 'that relate ranges of instruction counter values to ... exception handling').

Regarding Claim 40: The rejection of claim 39 is incorporated; further Schilling discloses the table indicating instruction identifiers for instructions that cause faults and

identifiers for the handling code units (pg. 40, col. 1, par. 5 'that relate ranges of instruction counter values to ... exception handling').

Response to Arguments

Applicant's arguments in the paragraph bridging pgs. 8 and 9 have been considered but are moot in view of the new ground(s) of rejection.

Applicant's arguments in the second paragraph on pg. 9 have been fully considered but they are not persuasive.

In the second paragraph on pg. 9, Applicant states:

> Frequency of execution of a null pointer condition check does not indicate how frequently the check encounters null pointer conditions.

With respect, Examiner believes Applicant misunderstood the indented combination. Examiner's position is that the teachings of Benitez be used to monitor the path taken as a result of the null pointer condition check, not the execution of the check it's self, and that this would provide an indication of the frequency of null pointer conditions.

In the same paragraph, Applicant goes on to state:

> There is no evidentiary support that specifically suggest modifying Kawahito to utilize the hot trace of Benitez

Examiner respectfully disagrees. The evidentiary support is provided by Benitez's teachings from col. 2, lines 28-31 'selecting sequences ... that are most amenable to dynamic optimization', as indicated in the previous action, and maintained here. As further support for the combination Examiner points to Benitez's abstract 'reduces

unnecessary translations and optimizations, and thereby increases execution speed and reduces the usage of memory and other resources' and further to col. 1, lines 28-31 'The invention generally relates to ... dynamic translating compilers' which includes Kawahito's invention, as indicated in the first action (pg. 139, col. 1, par. 1 'written in JAVA').

Further Applicant's arguments fail to comply with 37 CFR 1.111(b) because they amount to a general allegation that the claims define a patentable invention without specifically pointing out any deficiencies in the motivation to combine.

Applicant goes on to state:

> The modification or combination would still fail to disclose or suggest particularly eliminating null pointer condition checks that infrequently encounter null pointer conditions.

Examiner respectfully disagrees, on the grounds stated in the first action, which are maintained here, and for the reasons stated above regarding Applicant's arguments in the second paragraph on pg. 9.

Conclusion

The prior art made of record and not relied upon is considered pertinent to applicant's disclosure. US 6,158,049 to Goodwin et al. and US 6,631,518 to Bortnikov et al. both disclose optimization based on profile feedback.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Jason Mitchell whose telephone number is (571)

272-3728. The examiner can normally be reached on Monday-Thursday and alternate Fridays 7:30-5:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kakali Chaki can be reached on (571) 272-3719. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Jason Mitchell
4/20/05

TODD INGBERG
PRIMARY EXAMINER